Hierarchical Cluster Language Modeling For Rescoring N-Best Hypotheses During Speech Decoding

Photina Jaeyun Jang, Alexander G. Hauptmann
{jaeyun, alex}.cs.cmu.edu

Computer Science Department
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
U.S.A.

ABSTRACT

We propose an unsupervised learning algorithm that learns hierarchical patterns of word sequences in spoken language utterances. It extracts cluster rules from training data based on high n-gram probabilities to cluster words or segment a sentence. Cluster trees, similar to parse trees, are constructed from the learned cluster rules. Through hierarchical clustering we are adding grammatical structure onto the traditional trigram language model. The learned cluster rules are used to improve the n-best utterance hypothesis list which is output by the Sphinx III speech recognizer. Our hierarchical cluster language model is used to rescore and filter these n-best utterance hypotheses. It assigns confidence scores to segments of hypotheses that can be clustered hierarchically with the learned cluster rules. Rescoring the original n-best hypothesis list, which is based on acoustic and trigram language model scores, with our hierarchical cluster language model results in a set of hypotheses with lower word error rate. Our cluster language model was trained on TREC broadcast news data from 1995 and 1996, and tested on the HUB-4 '97 development test broadcast news data. Compared to manually created grammar rules, the cluster trees more accurately reflect the speech data since their cluster rules are automatically learned based on empirical n-gram probabilities from the training data, whereas manually written grammar rules can introduce human bias, and are expensive to develop. Prior symbolic knowledge in the form of rules can also be incorporated by simply applying the rules to the training data before the earliest applicable learning iteration. Our algorithm is also able to learn clusters reflecting various styles of data: whether the language is formal, strictly grammatical or loose conversational speech.

1. INTRODUCTION

Stochastic language modeling approaches have been popular, employing maximum entropy, mutual information, information gain, maximum likelihood, expectation maximization, etc. [2,4,5]. Our hierarchical clustering approach using the top n-grams and their probabilities has an additional advantage with its white box approach: compared to the purely stochastic approaches, the discrete symbolic clusters which correspond to symbolic rules are visible at any learning step. This visibility aids in optimizing the parameters of our model and also allows symbolic knowledge to be inserted to help guide the empirical learning.

Traditionally, parsing in speech systems has been mostly applied after speech decoding for further speech understanding. We show how shallow parsing applied to the n-best hypotheses list during speech decoding can help improve the recognition accuracy.

2. HIERARCHICAL CLUSTERING

Our assumption is that the higher the probability of a certain n-gram within the training data, the more likely it is to occur again together as a sequence. We group those words or clusters within that n-gram as one cluster hierarchically, and consider that cluster or hierarchy of clusters as a valid utterance.

In contrast to approaches which treat a cluster as a bag of words, we will define a cluster as one n-gram of words or a (sub)tree constructed from previously clustered n-grams. We are using n-grams to keep the sequence of patterns, for example the left to right word order which applies to many languages. We also interpret a cluster as either a higher level class or as a rule. If a cluster is viewed as a class then it is defined by the unordered set of its component elements. If the cluster is a rule, we define the cluster name as the head of the rule and the elements of the cluster, i.e. the n-gram, constitutes the body of the rule corresponding to the left to right order of the literals, respectively.

In the example of Figure 1 below, the NP cluster is a class consisting of the elements NP1, NP2 and NP3, where the doesn't matter. The clusters NP1, NP2 and NP3 are examples of rules, where the order of their elements becomes important as for example in $NP3 \leftarrow Det \cdot Adj \cdot N$.

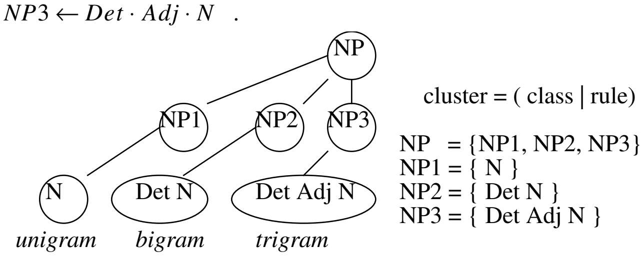

Figure 1: N-gram clusters are defined as either a class or a rule. NP is a cluster class, while NP1, NP2 and NP3 are cluster rules.

2.1. Algorithm

The steps to learn our hierarchical cluster language model are :

1. Build a n-gram language model which is a list of n-grams associated with their probabilities $p\langle w_n | w_1 w_2 \ldots w_{n-1} \rangle$
2. Select the top n-grams by their sorted n-gram probabilities. Each n-gram becomes one cluster (a cluster with *n* elements) and is assigned a cluster id.
3. The cluster id of each cluster is substituted back into the training data.
4. Iterate the above steps until each sentence in the original training data has been replaced by one cluster label.

It is important that the cluster id of the newly learned cluster, not the n-gram itself, is substituted back into the training data so that the algorithm can construct the next higher level of the cluster tree. We view a parse tree from bottom up as a sequence of words clustered into multiple higher levels. A cluster (sub)tree can be *at most* as high as the number of training iterations so far. Often the trees are shorter since not every previously clustered node is selected for clustering in the next iteration based on its probability. All sentences in the training data are guaranteed to be shallow parsed with the learned cluster rules.

If the unit for terminating the algorithm is not restricted to complete sentences, but relaxed to also allow terminating at phrases or segments, the learning step doesn't need to iterate until each sentence is replaced by a single cluster, but only until a reasonable proportion of the training data has been clustered. The termination of the learning can be done manually by examining the current cluster rules, which is one of the "white box"advantages of our approach.

Backing off by cluster subtrees would be analogous to a n-gram language model that backs off to n-1.

2.2. Properties of the Algorithm

Since n-grams are constructed statistically, many types of data can be used by our approach. We have learned hierarchical cluster language models with different character fonts or languages such as Korean or English, from either lexical text or part-of-speech (POS) tags. The style of the language, i.e. whether it is strictly grammatical or more conversational can also be learned without changing the learning algorithm as long as the training data contains representative examples of that style. This is another advantage since manually constructed grammars for a very loose conversational language style may be more difficult for humans to write. Also, human bias about grammar construction can be eliminated by learning purely empirically and statistically from training data. On the other hand, certain clusters can be forced to be learned prior to our learning process by clustering and substituting those prior rules into the training data before the training process. The head of the prior rule will correspond to a cluster id, and the body of the rule is the n-gram to be clustered and substituted by that cluster id, before the n-gram language model is built during the first step of the training algorithm. If we have prior knowledge of symbolic rules at intermediate levels of a subtree, the prior rules will be inserted into the clustered training data at the earliest applicable iteration. Also, if our training data does not contain sentence boundaries, or if we choose to remove the termination constraint at the sentence units, our learning algorithm has the inbuilt capability to learn clusters or rules across sentences which is useful for speech recognition of broadcast news streams.

To make the learning from very large corpora more tractable and also to add more generalization power to the cluster language model, the lexical training data was tagged with Brill's POS tags [1]. Since our algorithm is unsupervised learning, clusters with POS tag errors will not disable the learning procedure, but the cluster tree will merely be constructed differently. Once the text data is tagged, the POS tags that correspond to cluster labels can be more straightforwardly mapped into grammar variables. At this point, head-rules (XP $\leftarrow$ ... X ... , where X = N, V, A, P) can be applied as prior knowledge to learn more linguistical grammar rules. Preliminary work indicates that constraining the cluster language model to learn only bigram rules leads to more conventional parse trees.

2.3. Examples

Figure 2 shows an example of a cluster tree created for POS tagged text.

We denote the cluster rule and cluster id as the following:

cluster rule : cluster id $\leftarrow$ ngram
cluster_id = iteration . ngram_rank_id . ngram_size
ngram_rank_id = rank .-. subrank

For example, in the cluster_id '1c3', '1' denotes the 1st clustering iteration, 'c' represents 3rd ranked n-gram by its probability, and '3' denotes a trigram.

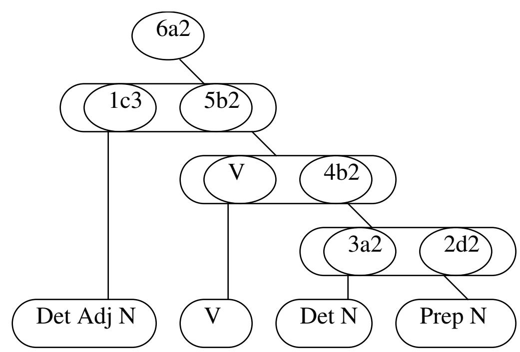

Figure 2: An Example showing a cluster tree for POS tagged text

To further illustrate our algorithm, we give examples of lexical text. Below is the initial training text before learning the cluster rules:

well the state department has said all along its bottom line is to contain saddam hussein but now u. s. officials are under pressure c. n. n.'s steve hurst is at the ...

With prior knowledge, like, for example knowledge about compound nouns, the training text would look initially as follows:

well the (state department) has said all along its (bottom line) is to contain (saddam hussein) but now (u. s. officials) are under pressure c. n. n.'s(steve hurst) is at the ...

The training text substituted with cluster ids after the 5th iteration looking up the cluster ids shown in Figure 3:

well 5.25-69.2 said all 5.25-2387.2 5.25-1667.2 to 5.25-1945.2 now 1.8-1.2 officials are under pressure

The first column in Figure 3 shows the cluster ids, the second column their log n-gram probability, the cluster label xp is a placeholder for the head of the cluster rule. If the training data consists of POS tags, the placeholder is replaced by the grammar variable which corresponds to the head of the parse rule. The grammar variable is obtained by the prior knowledge of head rules. This grammar variable will become a member of a *class* as explained in Figure 1, not a literal of a cluster rule learned from the next level clustering. If a lexicon is available containing lexemes along with their POS tags, lexical text can be mapped into corresponding POS tagged text. The lexicon is used as the prior knowledge, the lexeme to POS mapping rule is applied to the text prior to the learning iterations. The 4th column shows the n-gram which is clustered. The parse for the cluster is shown as the segmented cluster with its lexically rewritten cluster id *within* the n-gram. Note that the cluster_id in the first column is the higher level node for the cluster tree.

2.2-1.3 -0.0859 xp 1.43-1.2 *your call*
(*thanks for*) *your call*
4.1-1.2 -0.1579 xp 3.3-2.2 2.16-2.3
((*we should not*) *be*) (*a nation that*)
4.3-1.2 -0.2879 xp 3.24-1.2 *much*
((*thank you*) *very*) *much*
5.25-69.2 -0.7172 xp 4.43-8.3 has
(the state department) has
5.25-1667.2 -0.7172 xp 4.48-290.2 is
(bottom line) is
5.25-1945.2 -0.7172 xp 4.48-561.2 but
(contain (saddam hussein)) but
5.25-2387.2 -0.7172 xp along its

Figure 3: Example of Cluster Rules with cluster ids, log probabilities and corresponding n-grams.

3. SPEECH DECODING

3.1. Sphinx III decoder

The SphinxIII speech recognizer [6] first decodes with a forward Viterbi beam search using continuous density acoustic models and thereby produces a word lattice for each segment. For the best path search, a word graph is constructed from the lattice to search for the global best path according to a trigram language model and an empirically determined optimal language weight using a shortest path graph search algorithm [3]. To generate n-best lists for each segment, A* search is applied to the lattices produced by the Viterbi beam search. Two passes are used to get the initial recognition. The first pass is a conventional beam search using the Viterbi algorithm. It produces a word lattice that includes word segementations and acoustic likelihoods. The second pass is an A* search through a word graph constructed from the word lattice. The top of the nbest list from this search is the final recognition hypothesis.

3.2 Rescoring of n-best Hypotheses

Each clustered segment within the hypotheses is assigned a confidence score based on its *total n-gram probability and its n-best rank* computed from *the cluster_ids that construct the cluster tree*.

Confidence scores are only assigned for segments of hypotheses where cluster trees are *higher than level two*, i.e. segments clustered twice or more. We ignored single clusters since they would correspond to a n-gram, thus in case of a traditional flat trigram language model, a trigram would already contain the cluster information of the three word cluster. Computing the confidence score for cluster trees higher than two:

Multiple clusters which have the *same* associated n-gram probabilities within one iteration, are ranked *equally*, which is often the case during the learning procedure. The ngram_rank_id is rewritten with an additional subrank to discriminate between those clusters. Multiple clusters with same n-gram probability and same n-gram rank with different subranks, cause the subtrees to be clustered more rapidly.

ngram_rank_id = rank .-. subrank

The value of the subrank is assigned based on alphabetical order of the n-gram. For example, the ngram_rank_id: 2-4 denotes the 4th 2nd best cluster alphabetically ordered. All the clusters with rank 2-n have the same probabilities.

Many valid word sequences within n-best hypotheses can be found and verified already at the second or third level of shallow parsing if they match their corresponding cluster rules of our language model. Therefore it is not always necessary to iterate our learning algorithm until the topmost sentence node.

3.3 Example

The following example contains the original top two hypotheses produced by the Sphinx III speech decoder.

1: <s> *oh santa barbara said all along its bottom line is contain saddam hussein but now u._s. officials are under pressure c._n._n.'s steve hurst at_the state department store and_he joins us now with latest steve is pressures in* </s>

2: <s> *oh santa barbara said all along its bottom line is contain saddam hussein but now u._s. officials are under pressure c._n._n.'s steve hurst at_the state department store and_he joins us now with latest steve this pressures in* </s>

Below are the corresponding top two hypotheses substituted with the cluster ids based on the cluster rules.

1: *oh santa barbara said all 6.17-1515.2 8.11-1833.2 9.3-6761.3 2.6-4.3 9.3-1075.3 department store and he 5.35-4.2 with latest steve is pressures in*

2: *oh santa barbara said all 6.17-1515.2 8.11-1833.2 9.3-6761.3* 2.6-4.3 *9.3-1075.3 department store and he 5.35-4.2 with latest steve this pressures in*

The two best hypotheses segmented by clusters to visualize the parse:

1: *oh santa barbara said all ((along its) ((bottom line) is)) (((((contain (saddam hussein)) but) now) (u. s.)) officials) (are under pressure) (c. n. n.'s) ((steve hurst) (at the) state) department store and he ((joins us) now) with latest steve is pressures in*

2: *oh santa barbara said all ((along its) ((bottom line) is)) (((((contain (saddam hussein)) but) now) (u. s.)) officials) (are under pressure) (c. n. n.'s) ((steve hurst) (at the) state) department store and he ((joins us) now) with latest steve this pressures in*

Cluster rules that were learned at earlier iterations imply more confidently decoded sequences of words since it assures that the expressions are frequently used in that language.

4. EXPERIMENTS

The cluster language model was trained on HUB-4 's 1997 broadcast news development set. The cluster rules were learned with the *ngram_rank* parameter set to each 10-best, 20-best, 30-best , 40-best and 50-best. The cluster trees were built with 9 iterations. The n-best hypothesis list for N=500 was rescored with our confidence measure. The n-best list was filtered through our hierarchical cluster language model and confidence scores were assigned on the word segments that were parsed. After rescoring the n-best hypothesis list, the new highest scoring hypothesis becomes the final system output. As a result, we obtained 3.8% absolute decrease in word error rate on the HUB-4 1997 development set.

In a separate experiment, we clustered the top 10-best hypotheses of each utterance in the HUB-4 1997 development set with the learned cluster language model from TREC broadcast news data from 1995 to 1996. The union set of cluster rules learned with the ngram_rank parameter again set to 10-best, 20-best, 30-best, 40-best and 50-best constructed our new hierarchical cluster language model. Then the n-best list was rescored with the confidence scores of each clustered segment of the hypotheses to select the top 3 best scoring hypotheses. The new *set* of 3-best hypotheses list resulted in a *decrease* in word error rate of 2.6% (overall WER 32.8%) compared to the original set of 3-best hypotheses list scored using acoustic and trigram language model scores only (overall WER 35.4%). The overall WER for the list of top best scoring hypotheses resulted in a *decrease* of 0.3 % absolute (overall WER 35.1%).

We expect that these results could be further improved based on different thresholding of the following parameters: n-gram size, rank cutoff, n-best threshold parameter, and number of iterations.

5. ONGOING WORK

For a more general experiment, a cluster language model was trained on TREC broadcast news data from 1992 to 1996. Although it is very practical to learn the clusters automatically, the massive amount of lexical training data results in an exhaustive list of cluster rules that is eventually a list of cluster n-grams in addition to the n-gram language model. For this data, we decided to pretag the lexical training data rather than to use more lexical data for broader coverage.

To make learning on large corpora more tractable and also to add more generalization power to the cluster language model, the lexical training data was tagged with Brill's POS tags[1]. Once the text data is tagged and clustered, the POS tags that correspond to cluster labels can be interpreted as grammar variables of a grammar rule head or as the literals constituting the body of grammar rules. Alternatively, after tagging the training data with POS tags, simple head-rules using these tags can be applied as prior-knowledge clusters to aid in learning more classical grammars. Constraining the cluster language model to learn only bigram rules also leads to more conventional parse trees. However, we have not yet completely evaluated the quality of these parse trees.

Currently, during one iteration, multiple clusters of equal rank are *not* substituted *simultaneously* to create multiple (disjunctive) cluster paths, but are substituted sequentially in alphabetical order. We will have to investigate how much this affects the final set of cluster rules learned with different n-gram rank cutoff parameters.

6. DISCUSSION

We expect a more general set of rules to be learned when the set of rules learned with *different n*'s of the n-gram language model is merged into to a single *set* of rules.

In ambiguous parses, the partially clustered n-best hypothesis can be further bottom-up parsed by searching one level further within the list of cluster rules. Multiple ambiguous parses on the same length of the segments have not been examined yet.

Rescoring of the n-best hypothesis list was done so far using the lexical words only, but without making use of the available acoustic scores and language model scores of the Sphinx III system. Augmenting the N-best rescoring algorithm with the acoustic score provided by the Viterbi decoder and the language model score should be another straightforward step.

Applying the cluster language model not on the n-best hypothesis list but on the Viterbi lattice before the n-best list is generated, would prune invalid subsegments within the lattice and therefore speed up the A* search to produce the n-best list resulting also in a shorter n-best list.

REFERENCES

1. Brill, E., "*Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging*", Computational Linguistics, Vol. 21, No. 4, December 1995.

2. McCandless, M.K., Glass, J.R., "*Empirical Acquisition of lan guage models for speech recognition*", International Confer ence on Spoken Language Processing (ICSLP) 1994.

3. Ravishankar, M., "*Efficient Algorithms for Speech Recogni tion*", PhD. thesis, Carnegie Mellon University, Computer Science Dept. tech report CMU-CS-96-143, 1996.

4. Ries, K., "*A Class Based Approach To Domain Adaptation And Constraint Integration For Empirical M-Gram Models*" Proceedings of Eurospeech 1997.

5. Ries, K., Buo, F.D., Wang, Y., "*Improved Language Modeling By Unsupervised Aquisition of Structure*", International Conference on Acoustics Speech and Signal Processing (ICASSP) 1995.

6. Seymore, K., et al "*The 1997 CMU Sphinx-3 EnglishBroad cast News Transcription System*", Proceedings of the 1998 DARPA Speech Recognition Workshop 1998.